Spectral Techniques in Digital Logic

S. L. HURST

D. M. MILLER

I. C. MUZIO

Spectral Techniques in Digital Logic

S. L. HURST
University of Bath
Bath, England

D. M. MILLER
University of Manitoba
Winnipeg, Manitoba

J. C. MUZIO
University of Victoria
Victoria, British Columbia

1985

ACADEMIC PRESS
(Harcourt Brace Jovanovich, Publi[redacted]ers)

London Orlando San Diego [redacted]
Toronto Montreal Sydney Tokyo

ACADEMIC PRESS INC. (LONDON) LTD.
24–28 Oval Road
LONDON NW1 7DX

United States Edition published by
ACADEMIC PRESS, INC.
Orlando, Florida 32887

British Library Cataloguing in Publication Data

Hurst, S. L.
Spectral techniques in digital logic.
1. Electronic digital computers 2. Logic design
I. Title II. Miller, D. M. III. Muzio, J. C.
621.3819'582 TK7888.3

Library of Congress Cataloging in Publication Data

Hurst, S. L. (Stanley Leonard)
Spectral techniques in digital logic.

Bibliography: p.
Includes index.
1. Logic design. I. Miller, D. M. II. Muzio, Jon C.
III. Title.
TK7868.L6H88 1985 621.3819'5835 83-83427
ISBN 0-12-362680-3 (alk. paper)

PRINTED IN THE UNITED STATES OF AMERICA

85 86 87 88 9 8 7 6 5 4 3 2 1

Preface

During the past two decades a new field of digital logic theory has been born and its boundaries continuously extended. Two motives for this work may be discerned: first, the purely academic one of applying existing mathematical techniques to pastures new, and second, the increasing appreciation of the limitations of existing algebraic and geometric methods in handling digital data for logic network design purposes.

Digital logic design is a peculiar discipline in that the logic design for a given requirement, which may be in total a very large system, is frequently made without the use of any sophisticated design procedure. Very great sophistication, however, may be present in the logical verification of the complete assembly and in its translation into a microelectronic realisation, but the basic logic formulation may remain largely a hand-assembled synthesis, albeit aided by the experience of the logic designer who may be very proficient in intuitively recognizing patterns or symmetries in the network being designed. Indeed, many industrial designers will state that this is a perfectly acceptable situation, but this may be because of more demanding pressures from the existing areas of sophistication. Equally, the increasing capabilities of LSI and VLSI fabrication have so far kept pace with the designers' need, and no strong pressure for "better" design techniques has arisen.

However, the new field of digital logic theory represents a modern approach to expressing conventional digital data, one which can provide various insights into the structure of the data which are absent from classical Boolean algebra and truth-table formats. The pioneering work, particularly that of Karpovsky, originally in Russia, and Lechner, in the United States of America, forms the basis of this approach. Others have contributed and amplified the basic concepts and have translated the underlying mathematics into engineering tools which may be more acceptable to a digital logic designer.

The first chapters of this book will attempt to introduce the underlying theory of this area, that of orthogonal transforms and resulting spectral data. We assume that the reader will be conversant with conventional digital logic theory, such as is contained in any graduate textbook. Subsequent chapters will be concerned with the application of spectral data to Boolean function classification, logic network synthesis, fault diagnosis, and other aspects relevant to digital logic design. While the underlying theory of this area is applicable to any-valued logic, and not exclusively to the binary case, we will generally confine our discussions herewith to two-valued digital networks. Nevertheless, it is particularly significant that should future technology adopt a higher-valued logic, say quaternary, then the basic design techniques such as those discussed here will be applicable for such developments.

We hope that this book will be of interest to all working in computer engineering and digital system design, as well as to academic and research establishments. Our final hope is that it may maintain and increase interest in this area, leading to yet further developments which must surely be forthcoming.

S.L.H.
D.M.M
J.C.M.

Acknowledgements

The authors would like to express their thanks and appreciation for the work of others who have initiated and contributed to the present state of knowledge and application of spectral methods in digital logic design. Many have been referenced in the several chapters of this text, and we hope that this will be taken as our thanks to them for their work and inspiration.

The authors must also acknowledge financial support received from many sources in furthering their own work in this area and that of the research students with whom they have been associated. Among these benefactors we thank the United Kingdom Science and Engineering Research Council (SERC) for research studentships at Bath and the Natural Sciences and Engineering Research Council of Canada (NSERC) for their support of the work done in Canada.

In addition the SERC, the NSERC, the British Council, and particularly NATO have provided means for the authors to co-operate across the Atlantic for a number of years.

Finally, our thanks go to many colleagues in several countries for their personal comments and guidance in this still-evolving area of digital logic design.

S.L.H.
D.M.M.
J.C.M.

List of Symbols

a	Integer threshold-gate input weight
b	Entries of **B** (see below)
d	Integer threshold-gate output discrimination (threshold) value
e	Parameter used in fault detection procedures only
f	Function of all the input variables
g, h	Decompositions of a function
i	Independent input variable subscripts, $i = 1$ to n
j	Row subscripts of a matrix, $j = 0$ to $2^n - 1$
k	Column subscripts of a matrix, $k = 0$ to $2^n - 1$
m	Minterm identification with decimal notation subscript, see tabulation, also used in Fig. 3.8

m	x_n	x_{n-1}	$\cdots$	x_2	x_1
m_0	0	0	$\cdots$	0	0
m_1	0	0	$\cdots$	0	1
m_2	0	0	$\cdots$	1	0
$\vdots$					
m_j					
$\vdots$					
m_{2^n-1}	1	1	$\cdots$	1	1

Note that x_n = most-significant digit and x_1 = least-significant digit; see also Append. A, Section 1.

n	Number of independent input variables
p	Number of multi-valued logic levels, $p \geq 2$, $p = 2$ in binary
r	Entries of **R**
s	Entries of **S**
t	Entries of **T**, or as a superscript the transpose of a vector or matrix
u	Sub-function identifier
x	Independent input variable x_i, $i = 1$ to n, x_1 least significant
y	Entries of **Y**, except locally in Figs. 3.1 and 3.8
z	Entries of **Z**
X	$\{x_1, \ldots, x_n\}$
$f(X)$	$f(x_1, \ldots, x_n)$
B	Vector of $n + 1$ modified Chow parameters
N	2^n
Z	Truth-table column vector for function $f(x)$, entries $\in\langle 0, 1\rangle$, in truth-table order m_0, $m_1, \ldots, m_{2^n-1}$ unless otherwise stated

Y Truth-table column vector for function $f(x)$, entries $\in\langle +1, -1\rangle$, in truth-table order $m_0, m_1, \ldots, m_{2^n-1}$ unless otherwise stated
T Any transform
T^n $2^n \times 2^n$ transform
(Note that all **T**^{n} transforms will be in Hadamard ordering unless otherwise stated.)
T^n_p $p^n \times p^n$ transform matrix (used principally in Section 2.7)
T$^n_{j\cdot}$ jth row of **T**^{n} (row vector), $j = 0$ to $2^n - 1$
T$^n_{\cdot k}$ kth column of **T**^{n} (column vector), $k = 0$ to $2^n - 1$
t_{jk} jth row, kth column entry of **T**^{n}
R Resultant spectrum of function **Z**, **Z** $\in\langle 0, 1\rangle$
S Resultant spectrum of function **Y**, **Y** $\in\langle +1, -1\rangle$
r_j jth entry of column vector **R**, $j = 0$ to $2^n - 1$
s_j jth entry of column vector **S**, $j = 0$ to $2^n - 1$
I^n $2^n \times 2^n$ identity matrix (n dropped where no ambiguity arises)
J^n 2^n column vector whose top entry takes the value 2^n, all remaining entries zero-valued
β $2^{n-m} - 1$, where m is $1 \le m \le n$, see sub-function definitions
$\bar{}$ Used above a binary variable or function to indicate the complement (negation)

General Transform of a Function $f(X)$

$$\mathbf{TY} = \mathbf{S}, \quad \mathbf{Y}\in\langle +1, -1\rangle, \qquad \text{or} \qquad \mathbf{TZ} = \mathbf{R}, \quad \mathbf{Z}\in\langle 0, 1\rangle$$

Sub-functions of a Binary Function $f(X)$

$f_u(x_1, \ldots, x_m) \equiv f(x_1, \ldots, x_m, u_1, \ldots, u_{n-m})$, where u is a constant and $(u_1, \ldots, u_{n-m})$ is the binary expansion of u, i.e., $u = \sum_{i=1}^{n-m} u_i 2^{i-1}$
Y$_u$, **Z**$_u$ minterm column vectors for f_u, $Y_u \in\langle +1, -1\rangle$, $Z_u \in\langle 0, 1\rangle$, respectively
S$_u$, **R**$_u$ spectrum for **Y**$_u$, **Z**$_u$, respectively

Decomposition g, h of a Function $f(X)$ (see Fig. 4.3)

$f(X) = h(X, g(X))$, where g, h may be any function of n input variables
R^g, **S**^{g} the spectrum of function g in the above decomposition
$\hat{\mathbf{R}}$, $\hat{\mathbf{S}}$ the spectrum of function h in the above decomposition

Additional Superscript Notations

- $\tilde{}$ In Chapters 4 and 5 used above an x_i input variable when it is a remapping from the original x_i input variables, for example, $\tilde{x}_2$
- $\hat{}$ In Chapter 5, used above a function or spectral parameter where the given function $f(X)$ contains undefined ("don't-care") minterms, and when the don't-care minterms are all allocated the logic value 0, for example, $\hat{s}_3$
- $\hat{\hat{}}$ In Chapter 5, as above but when all the don't-care minterms are allocated the logic value 1, for example, $\hat{\hat{s}}_3$
- $*$ In Chapter 6, used above a function or spectral parameter to indicate that it is a parameter of a faulty function, for example, $\overset{*}{s}_1$

Mathematical Symbols

$+$ Arithmetic addition, or maximum of, = Boolean addition for $m = 2$ case, = logical OR (the context of use should identify which meaning is present)
$\times$ Arithmetic multiplication
Minimum of, = Boolean multiplication for the $p = 2$ case, = logical AND (symbol dropped where no ambiguity occurs)

$*$	Convolution
$\circledast$	Kronecker matrix product
$\oplus$	Addition mod_p, = Exclusive-OR for $p = 2$ case
$\otimes$	Multiplication or product mod_p
$\bar{}$	Cyclic negation, = NOT for the $p = 2$ case
j	$\sqrt{-1}$, $= 1.0 \angle 90°$
a	$\exp(2\pi j/3) = -0.866 + j0.5 = 1.0 \angle 120°$; used principally in Section 2.7
b	$\exp(2\pi j/p) = 1.0 \angle 360°/p$; used principally in Section 2.7
$H_i^q, H_i^{q,r}$	Haar functions

Contents

1

General Introduction

1.1 BOOLEAN AND SPECTRAL DOMAINS

In this book we are concerned with digital logic and with the design and analysis of switching circuits. The majority of existing methods are concerned with the properties of Boolean functions since it was proved by Shannon[1] that the Boolean domain provides a precise model for the analysis of switching circuits. We are only concerned with a two-valued Boolean algebra for practical applications, with these two values normally being represented by 0, 1, irrespective of their actual implementation. The behaviour of a device is represented by a function $f(x_1, x_2, \ldots, x_n)$ of its input variables $x_1, \ldots, x_n$. This function can most conveniently be defined by a table. For example, a function $f(x_1, x_2, x_3)$ of the three variables x_1, x_2, x_3 is illustrated in Table 1.1.

It is common to use the product and sum operators of the Boolean algebra together with negation to define such functions—for example, $f(x_1, x_2, x_3) = x_1\bar{x}_2\bar{x}_3 + \bar{x}_1x_2\bar{x}_3 + x_1x_2x_3$. The use of Boolean algebra for the manipulation and analysis of switching circuits is well known and is not part of our purpose in this book. We shall be looking at a different domain in which to express the information required to define and analyse a Boolean function.

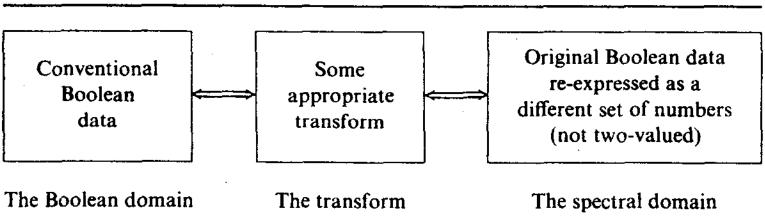
Part of the problem with the definition in the Boolean domain is that each of the entries in the column for $f(x_1, x_2, x_3)$ in Table 1.1 tells us precisely the behaviour of the function at a single point but nothing of its behaviour for any other points. It is possible to give an alternate representation of a function where the information about the function is much more global in nature. This alternate representation is in the spectral domain, and it will be demonstrated in the later chapters that a number of properties are much more easily deduced in the spectral domain than in the Boolean one.

The basic idea of the spectral domain, and how to get there, is illustrated in Table 1.2. If we are to avoid losing information, we shall have to ensure that the transform can be reversed, that is, that we can move to and from the spectral domain without any loss of information.

Table 1.1. The function $f(x_1, x_2, x_3) = x_1\bar{x}_2\bar{x}_3 + \bar{x}_1 x_2 \bar{x}_3 + x_1 x_2 x_3$.

x_3	x_2	x_1	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Table 1.2. The transform.



To transform the Boolean representation (such as that for $f(x_1, x_2, x_3)$ given in Table 1.1) into the spectral domain, we take the right-hand column from the defining table for a column vector **Z**. Using a particular square matrix **T** of the correct size as our transform gives the spectrum $\mathbf{R} = \mathbf{TZ}$ for the function. This vector **R** (of the same size as **Z**) is an alternate representation for the function as long as **T** has an inverse. For the particular transforms that we shall be using, the inverse is very straightforward. A detailed discussion of these is given in Chapter 2.

1.2 HISTORY

Here we shall only give a brief outline of the major developments that have led to the results that are described in the later chapters. The basis for the transforms goes back to Rademacher[2] and Walsh[3] and the transforms themselves are particular examples of Hadamard[4] matrices. Their early work was followed by others in the area of studying orthogonal functions.[5–8]

The question of the evaluation of the transform has been extensively studied, demonstrating that the evaluation can be limited to just additions and subtractions and moreover only involves $n \cdot 2^n$ operations for a $2^n \times 2^n$ transform.[9–12] These fast transform procedures are explained in Section 2.4.

The applications of orthogonal functions in the digital area came first in the areas of signal processing and the transmission of information. For a detailed discussion of this work, the reader is referred to Ahmed and Rao[13] and Harmuth.[14] In applications to digital logic and circuit design, analysis and synthesis, the first suggestion that orthogonal functions might be useful was made by Coleman.[15] He suggested that orthogonal functions might be useful for the design of circuits, and this theme recurs through much of Chapters 4 and 5. The appropriate references will be found in the chapters concerned. A number of books have addressed the use of orthogonal functions and Walsh functions in the digital area, and the reader is referred to them for related work in this area.[16–18]

Following the work of Dertouzos[19] and Lechner[20] it has been shown that spectral methods yield powerful classification techniques for functions (see Chapter 3). These classification methods can be used as the basis for synthesis algorithms and realization techniques.[21,22] They have also been used as a basis to design universal logic modules.[23] In the area of sequential machine design and state optimization some results have been reported.[24,25] Besslich[26,27] and Lloyd[28] have considered the idea of prime implicant extraction in the spectral domain and covering problems.

In 1955 Chrestenson[29] generalized the work of Walsh[3] to the many-valued case, that is, we are no longer transforming from the Boolean two-valued domain, but from a multiple-valued one (p-valued, $p \geq 2$). The procedure of using multiple-valued transforms is much more complex than two-valued and is considered briefly in Section 2.7, but is otherwise outside the scope of this book. Karpovsky[17] gives good coverage to multiple-valued results and the reader is also referred to the recent work of Moraga.[30–32]

This very brief summary is not intended to provide any detailed coverage for the various topics. This will be done as they are discussed in the later chapters. It does however illustrate some of the diversity of the areas of application of the spectral techniques.

1.3 MOTIVATION

The question arises as to the reason for considering the spectral domain and if there are any real purposes for its use. To understand the first difference between the Boolean and spectral domain, let us consider a Boolean function $f(X)$ of n variables. One row of the table defining this function provides complete and precise information about the behaviour of the function for one combination of the input variables. Of course, it does not tell us anything about the value of the function anywhere else. The combination of the knowledge of the behaviour of $f(X)$ for the 2^n rows of the table gives a complete definition of the function. Similarly, the spectrum for a function of n variables also contains 2^n values, which together completely define the function and can be used to recover its Boolean specification. Each of the 2^n values in the spectrum (the spectral coefficients) contains some information about the behaviour of the function at all 2^n points, but does not contain complete information about any of them. The combination of all the values in the spectrum does lead to complete information about the function, but each individual coefficient gives us some global information about the whole function. In this sense the spectral coefficients are giving us global information about the function, while the Boolean domain consists of local information. For some applications this global information is more directly useful than the Boolean representation of the function.

The easiest way to demonstrate the value of the spectral methods is to give a brief description of the areas that are covered in the rest of the book. Chapter 2 gives a complete explanation of the spectra of discrete functions and their calculation, together with some consideration to the use of other transformations using incomplete and non-orthogonal matrices.

The classification of Boolean functions is explored in Chapter 3, and an explanation is given of the way in which the spectral classification is connected to threshold functions and other properties of certain classes of functions. The synthesis and design of circuits is discussed in Chapters 4 and 5. A number of powerful techniques are described in Chapter 4, showing how certain properties of a function are easily detected in the spectral domain but may be very difficult in the Boolean domain. This enables a number of different approaches to design to be considered. Chapter 5 is entirely concerned with the detection of symmetries and partial symmetries in functions, and demonstrates their value in synthesis techniques.

The most recent application for spectral techniques has been in fault diagnosis. This exciting area is described in Chapter 6. One of the serious problems with digital circuits concerns their testing. There are two aspects to this—first to verify whether or not a circuit is performing correctly and, second, if there is a fault, to find its location (fault isolation). Of course, from the user's point of view there is not usually much purpose in locating a fault beyond identifying a chip that needs replacing. There is no need to identify the exact location of the fault on the chip. We shall be showing that a check of the correctness of certain spectral coefficients for a digital network ensures that the network is free of certain types of faults. The testing technique involved is straightforward and can be easily applied. The required coefficients for the circuit are called a signature, and the ease with which they can be derived using spectral techniques and the resulting high level of fault coverage give a number of promising new ideas for fault testing in digital circuits.

Therefore we hope to cover all aspects of the present state of the art of this subject area in the following chapters. Readers who may wish to be reminded of the properties of vectors and matrices, we refer to Appendix C in this text. Whilst this Appendix covers the general case, where the matrix entries may be complex numbers, in all our binary work we shall only need to be concerned with the particular case of real entries in the matrix operations. Complex entries become necessary only when higher-valued logic than binary is being considered, which is the subject of Section 2.7 only.

References

1. Shannon, C. E., A symbolic analysis of relay and switching circuits, *Trans. AIEE* **57**, 713–723 (1938).
2. Rademacher, H., Einige Sätze über Reihen von allgemeinen orthogonal funktionen, *Math. Ann.* **87**, 112–138 (1922).
3. Walsh, J. L., A closed set of orthogonal functions, *Amer. J. Math.* **45**, 5–24 (1923).
4. Wallis, J. S., "Hadamard Matrices" (Lecture Notes No. 292). Springer-Verlag, New York, 1972.
5. Paley, R. E. A. C., A remarkable series of orthogonal functions, *Proc. London Math. Soc.* **34**(2), 241–279 (1932).
6. Paley, R. E. A. C., On orthogonal matrices, *J. Math. Phys.* **12**, 311–320 (1933).
7. Fine, N. J., On the Walsh functions, *Trans. Amer. Math. Soc.* **65**, 372–414 (1949).
8. Selfridge, R. E., Generalized Walsh transforms, *Pacific J. Math.* **5**, 451–480 (1955).
9. Brown, R. D., A recursive algorithm for sequency-ordered fast Walsh transforms, *IEEE Trans. Comput.* **C-26**, 819–822. (1977).
10. Fino, B. J., and Algazi, V. R., Unified matrix treatment of the fast Walsh–Hadamard transform, *IEEE Trans. Comput.* **C-25**, 1142–1145.
11. Shanks, J., Computation of the fast Walsh–Fourier transform, *IEEE Trans.* **EC-18**, 457 459 (1969).
12. Ulman, L. J., Computation of the Hadamard transform and the R transform in ordered form, *IEEE Trans. Comput.* **EC-19**, 359–360 (1970).
13. Ahmed, N., and Rao, K. R., "Orthogonal Transforms for Digital Signal Processing." Springer-Verlag, New York, 1975.
14. Harmuth, H. F., "Transmission of Information by Orthogonal Functions." Springer-Verlag, New York, 1972.
15. Coleman, R. P., Orthogonal functions for the logical design of switching circuits, *IEEE Trans. Comput.* **EC-10**, 379–383 (1961).
16. Beauchamp, K. G., "Walsh Functions and Their Application." Academic Press, London, 1975.
17. Karpovsky, M. G., "Finite Orthogonal Series in the Design of Digital Devices." Wiley, New York, 1976.
18. Hurst, S. L., "The Logical Processing of Digital Signals." Crane-Russak, New York and Edward Arnold, London, 1978.